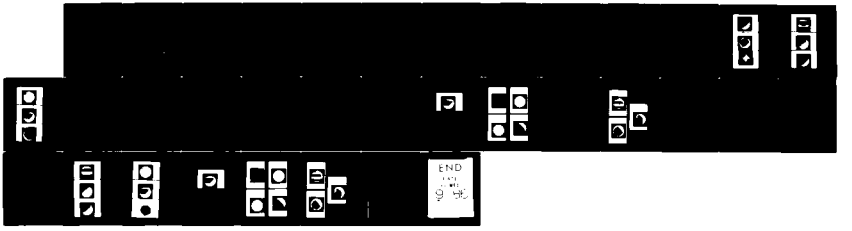
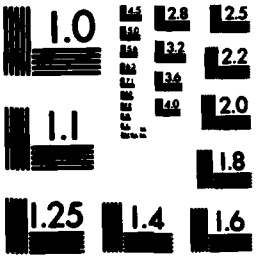
AD-A170 882 EXPERIMENTS IN ROTATIONAL EGOMOTION CALCULATION(U) 1/1
ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE B SARACHAN
FEB 85 TR-152 DACA76-85-C-0001

UNCLASSIFIED F/G 9/2 NL





Experiments in Rotational Egomotion Calculation

Brion Sarachan
Computer Science Department
University of Rochester
Rochester, NY 14627

TR 152
February 1985

14

Experiments in Rotational Egomotion Calculation

Brion Sarachan
Computer Science Department
University of Rochester
Rochester, NY 14627

TR 152
February 1985

Abstract

The motion of an observer relative to a fixed environment can be determined from a sequence of successive perspective images. The method is based on constraints between the motion parameters, the image intensities, and the shape of the body in view. A method is developed for experimentally verifying the algorithm using synthetic input for the case of rotational motion.

Technical problems arise in the representation of continuous mathematics in the discrete domain of a computer system. The effects of some of these difficulties are examined analytically and experimentally.

The preparation of this paper was supported in part by the Defense Advanced Research Projects Agency, U.S. Army Engineering Topographic Labs, under grant number DACA76-85-C-0001 and in part by the National Science Foundation under grant number DCR-8302038.

1. Rotational Egomotion Equations

This paper describes the implementation of the Egomotion Equation derived in previous work at the University of Rochester (Direct Processing of Curvilinear Sensor Motion from a Sequence of Perspective Images by Aloimonos and Brown, Feb. 1984). The idea is that a sequence of images obtained from a moving camera contains the information needed to obtain the motion parameters of the camera. The derivation is based on the Flow Constraint Equation (eq. 1).

$$f_x u + f_y v + f_t = 0. \qquad (1)$$

f_x, f_y, and f_t are the spatial and temporal derivatives of the image, and u and v represent the optical flow. It is assumed that the intensities of the images will be smoothly varying, except for some localized discontinuities such as boundaries. Therefore, most points of the image will have spatial derivatives which contain information as to the image intensities of their neighbors.

<fig. 1>

Equation (1) is applied to the geometric situation shown in (fig. 1). The object is stationary and the viewpoint is able to move with any constant velocity. A,B, and C represent rotational velocities about the viewpoint's axes, and U, V and W represent translational velocities. Aloimonos and Brown use the rotational and translational velocities and perspective relationships to derive the equations (eq. 2).

$$u = ((-U/Z) - B + Cy) - x((-W/Z) - Ay + Bx) \qquad (2)$$
$$v = ((-V/Z) - Cx + A) - y((-W/Z) - Ay + Bx).$$

Here, u and v are the optical flow, as before. (X,Y,Z) is an object point. (x,y) is the corresponding point on the image plane. Substituting (2) into (1) gives a relation for the velocitities in terms of image point coordinates and derivatives. If it is known in advance that the motion is purely rotational or purely translational, the unnecessary velocities can be set to zero.

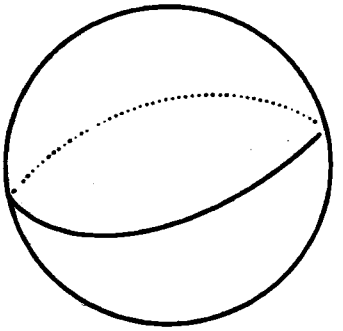
In deriving (2), the perspective relationships $x = X/Z$ and $y = Y/Z$ were used. These equations assume that the focal length is equal to one. In this implementation of the algorithm, the equations have been extended to include a variable focal length. Including focal length, the perspective relations become $x/f = X/Z$ and $y/f = Y/Z$.

Then, for purely rotational motion, setting U, V, and W equal to zero leads to the form of the Flow Constraint Equation shown in (eq. 3).

$$B(-f_x(x^2/f + f) - f_y x\, y/f) + A(f_x x\, y/f + f_y(y^2/f + f)) + C(f_x y - f_y x) + f_t = 0 \quad (3)$$

(Eq. 3) is the Egomotion Equation being verified in these experiments. Aloimonos and Brown verified the algorithm for the case of purely translational motion in their original paper.

The intuitive interpretation of (eq. 3) is that, given a point on the image plane (x,y), the spatial derivatives around this point compared with the the point's temporal



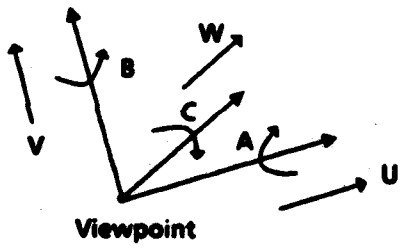


FIG. 1

derivative constrain the possible camera motions causing the changes. The perspective relations are involved in the equation, also, because the constraint obtained by the derivatives varies throughout the image plane because of the effect of perspective.

We now have an equation which, for any point of the image and its derivatives, has three unknowns, A,B, and C. Given any three points of the image, we could solve for these rotation velocities. Due to the effects of noise, though, and also the inherent approximations of representing continuous mathematics in the discrete structure of a computer system, the values obtained in sampling individual points will include some degree of error. It is best, then, to sample a large number of points and to use a statistical method to find the best answer for all of them. Two different methods are implemented here for comparison: least-squared-error fitting and mode-based fitting (the Hough Transformation).

2. Least-Squared-Error Method

The original paper by Aloimonos and Brown contains the closed form solutions needed in the implementation of this statistical method. The approach is to find an expression for the sum of the squares of the error for the set of points sampled. This expression is in terms of the motion parameters sought. Minimizing the expression will result in the most likely values for these motion parameters.

For the case of purely rotational motion, we must minimize the error given in (eq. 4).

$$\text{error}^2 = \iint (K(x,y)B + L(x,y)A + M(x,y)C + f_t)^2 \, dx\,dy. \qquad (4)$$

K,L, and M are the coefficient expressions of B, A, and C in the flow constraint equation. These expressions are slightly different from those listed in the original paper, because, again, it was assumed in the original paper that the focal length had a constant value of one. For these experiments, all equations are modified to allow for a variable focal length. K,L, and M are now:

$$\begin{aligned} K(x,y) &= -f_x(x^2/f + f) - (f_y x\, y)/f \qquad (5) \\ L(x,y) &= (f_x x\, y)/f + f_y(y^2/f + f) \\ M(x,y) &= -f_y x + f_x y, \end{aligned}$$

where f = focal length. (Eq. 4) is then minimized for a sample of points, and values are obtained for the motion parameters A, B, and C. All of the object points are sampled. Points whose grey levels correspond to the background color and whose adjacent points have grey levels corresponding to the background color are not used. In this way points on the background and points on the edge of the object are not included.

3. Hough Transformation Method

The Hough Transformation is an estimation method using the statistical mode. It can be used to obtain values for the motion parameters for comparison with least-squared-error fitting. In the Hough Transformation implementation, each sample point votes for all possible values of the motion parameters based on that

point's derivatives and the flow-constraint-equation. The votes are accumulated in an array, and the set of values with the most popular votes after all samples have been taken is the result.

The following algorithm is used for the rotational case:

```
given a three-dimensional accumulator array Acc[A, B, C]
for some number of sample points do
        choose a random point
        for A = Amin to Amax do
                for B = Bmin to Bmax do
                        if (f_x or f_y) and f_t > threshold do
                                solve for C
                                Acc[A, B, C] = Acc[A, B, C] + 1
```

Program macros define the size of the accumulator array, the incremental value between adjacent locations of the array, and offset values for A, B, and C. The program can thus be tailored to suit a given problem. For example, if the correct values of the rotation parameters are known in advance (or approximations of these values), the acculmulator array can be centered about these values. This allows there to be a large number of incrementally-close values in the array, while maintaining a reasonably-sized array in memory.

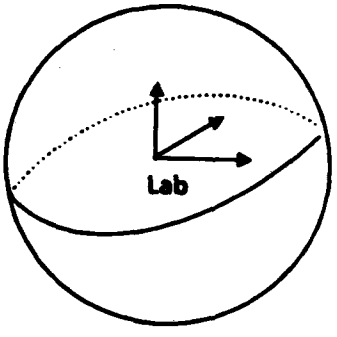
4. Camera Motion Model

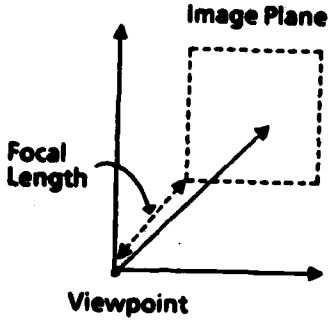
In order to test the algorithm, it was necessary to generate images using the proper geometry. A program written in 1983 by Linda Wilkins called "Sphere Movie" was modified for this purpose. The program uses ray tracing to create successive images of a moving sphere or cylinder in space. Any number of light sources can be positioned to illuminate the object. Also the object can have any of a number of surface reflectance functions and albedo patterns. However, the geometry of "Sphere Movie" was incorrect for this situation, so major modification was needed for the program.

As originally written, "Sphere Movie" defined its object with reference to a fixed three dimensional coordinate system and also moved the object with respect to this system. For the purpose of the egomotion algorithm, however, the camera rather than the object must experience motion defined by the motion parameters. "Sphere Movie" was modified so that the position of the camera is actually represented by the origin of a coordinate system (called the CAMERA). This coordinate system can be rotated and translated with respect to the object coordinate system (called LAB).

<fig. 2>

The origin of CAMERA, and the vectors representing the direction vectors of its axes, are defined in terms of points of LAB. The viewpoint is situated at the origin of the CAMERA, and the image plane is perpendicular to the CAMERA z-axis at a distance from the viewpoint equal to the focal length. The image is then created by ray-tracing.





CAMERA

FIG. 2

Both coordinate systems are represented by a 4x4 matrix:

$$\begin{vmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ c_1 & c_2 & c_3 & 0 \\ P_1 & P_2 & P_3 & 1 \end{vmatrix}$$

The point (P_1,P_2,P_3) is the location (in terms of LAB) of the center of the coordinate system. a, b, and c are unit direction vectors in the directions of the three axes. LAB, then, is represented simply by the unit matrix.

Aloimonos and Brown use the following equations for the instantaneous motion of the camera:

$$\begin{aligned} X' &= -U - BZ + CY \\ Y' &= -V - CY + AZ \\ Z' &= -W - AY + BX \end{aligned}$$

where (X',Y',Z') represents the velocity at (X,Y,Z).

In matrix form, this leads to a transformation matrix T:

$$T = \begin{vmatrix} 1 & -C & B & 0 \\ C & 1 & -A & 0 \\ -B & A & 1 & 0 \\ -U & -V & -W & 1 \end{vmatrix}$$

Because both the CAMERA and the LAB are defined in terms of LAB coordinates, multiplying either of these coordinate systems by T will transform it with respect to LAB coordinates. In other words, if CAMERA is represented by C and T is the transformation for a single rotation parameter, multiplying (C)(T) will lead to rotating C about an axis of LAB rather than a CAMERA axis.

In order to perform transformations on CAMERA with respect to CAMERA's own axes (i.e. to rotate CAMERA about its own axes and translate CAMERA in the direction of its own axes) the proper multiplication is $(C)(C^{-1})(T)(C)$. This first inverse transformation moves the CAMERA to coincide with LAB. The camera is then transformed by T, and then returned so that it will have been transformed relative to its original position. Since $(C)(C^{-1})$ is equal to the unit matrix, this leads to the simple result that the multiplication (T)(C) transforms the camera coordinate system with respect to its own axes.

It should be noted that the motion parameters A,B,C,U,V, and W as used in the egomotion algorithm represent instantaneous motion velocities. However, as used in the modified version of "Sphere Movie", these parameters generate finite, discrete motions.

If T were to represent a rotation about the z-axis, it would reduce to:

$$T = \begin{vmatrix} 1 & -C & 0 & 0 \\ C & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

The exact rotation matrix, T' would be:

$$T = \begin{vmatrix} \cos\beta & \sin\beta & 0 & 0 \\ -\sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

For small β, we can see that T' -> T, because $\cos\beta$ -> 1 and $\sin\beta$ -> 0 as β -> 0. A similar argument holds for a simple rotation about the x-axis or about the y-axis. However, for more complex transformations consisting of rotations about two or three of the axes combined with a translation, T becomes a poorer approximation of the precise transformation matrix. For example, consider a transformation that consists of the sequence of:

1) rotation about the x-axis by an angle c
2) rotation about the y-axis by an angle b
3) rotation about the z-axis by an angle a
4) translating by an amount x_T, y_T, z_T

The precise transformation matrix T' becomes:

$$T' = \begin{vmatrix} D & E & F & 0 \\ G & H & I & 0 \\ J & K & L & 0 \\ x_T & y_T & z_T & 1 \end{vmatrix}$$

where:

D = cosa cosb
E = sina cosb
F = -sinb
G = -sina cosc + cosa sinb sinc
H = cosa cosc + sina sinb sinc
I = cosb sinc
J = sina sinc + cosa sinb cosc
K = -cosa sinc + sina sinb cosc
L = cosb cosc

[Giloi, 1978].

T is a poorer approximation now to T' because each of A, B, and C must represent a pair of unequal matrix terms. For small angles, however, the approximation is still fairly close because each diagonal term of T' contains a term

which is a product of cosines, which will approach one for small angles. Each off-diagonal element of T' has terms that consist of products of sines with sines or products of sines with cosines. For small angles, these terms will be some small numbers which are approximated by A, B, and C in T. How close the approximation is depends on how close associated off-diagonal elements are to each other. For example, since A must approximate both I and -K, I and -K must be close to each other. Similarly, -F must be close to J, and E must be close to -G.

Repeated transformations by the exact transformation matrix, T, will retain the rigid camera geometry. However, transformations by the approximate transformation matrix, T', will result in a skewed camera coordinate system. The extent of this skew for the small angles used in these experiments and the effect that this has on the experimental results remains an open question.

5. Generating the Synthetic Images

The modified version of "Sphere Movie" has considerable flexibility. In generating an image of a sphere or a cylinder, a number of parameters can be controlled by the user. The size of the image can be specified. The radius of the object (as well as the height in the case of a cylinder) is specified in terms of "LAB space." Any number of light sources can be specified in terms of location and relative intensity.

An initial set of motion parameters is specified for the CAMERA in order to position the viewpoint and image plane in the proper location relative to the object. Another set of motion parameters is also specified to represent the incremental camera motion between generated image frames. (These are the parameters which are subsequently deduced by the egomotion algorithm.) The focal length of the camera is also specified.

Any of a number of options can be selected for the background pattern, the reflectance function of the object surface, and the albedo pattern drawn on the object surface (fig. 3). The number of "iterations," or the number of subsequent images to be generated, can also be set by the user.

All of this program input is stored in a file and is automatically loaded when the program is run. This saves the user from the time-consuming task of manually entering all the input each time the program is used. Only those input parameters which are to be changed need to be edited in the input file. The program itself is usually run as a background job because twenty minutes or more are required to generate a single image.

The ray tracing geometry is more complex in the modified "Sphere Movie" than in the original version. A visible point on the object is determined by finding the intersection of the line-of-sight with the object. A different line-of-sight passes through each point of the image plane, and each line-of-sight has the viewpoint as a common point. The object can be described by the analytic expression for a sphere or a cylinder.

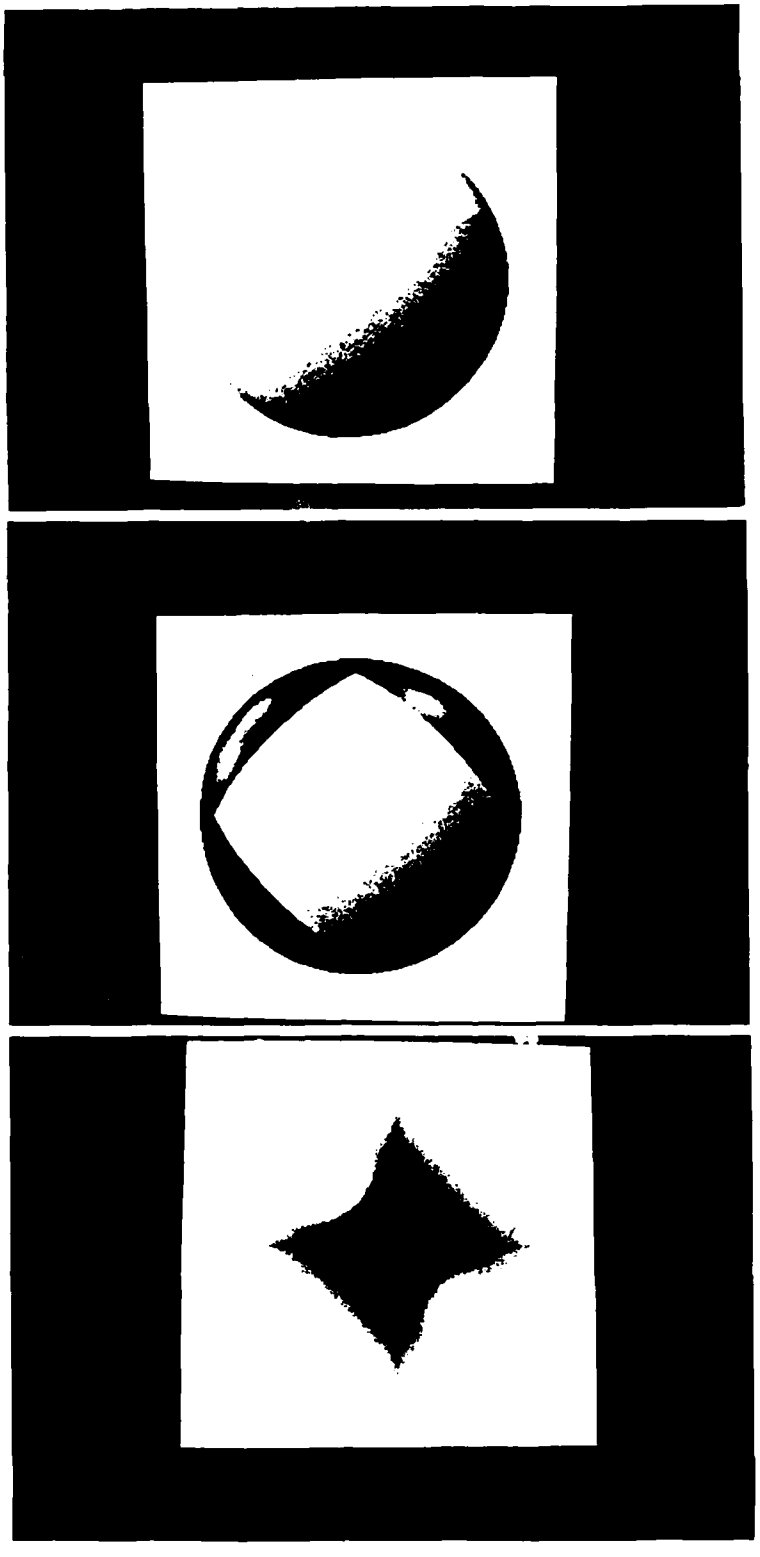


FIG. 3

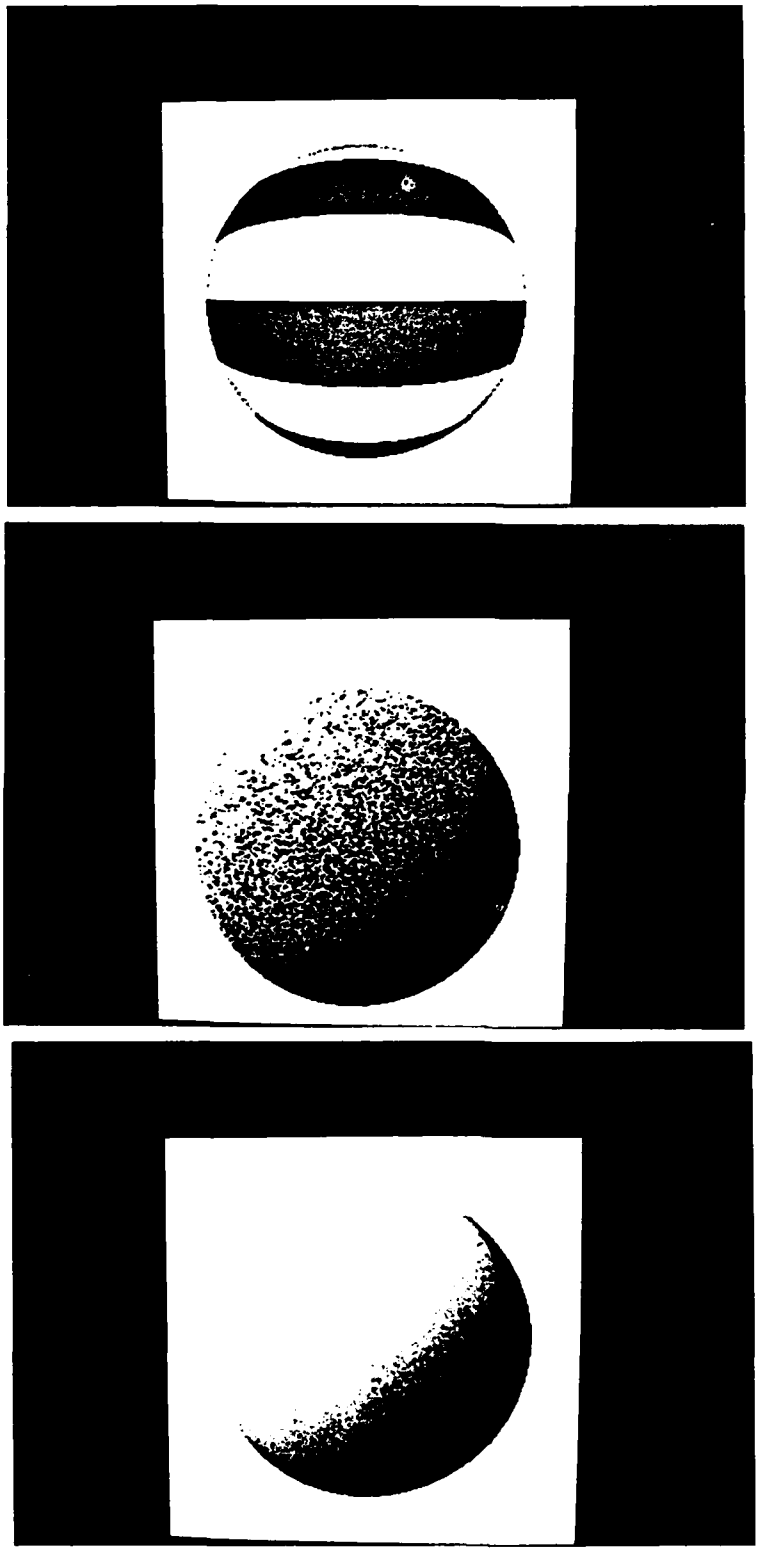


FIG. 3, CON'T.

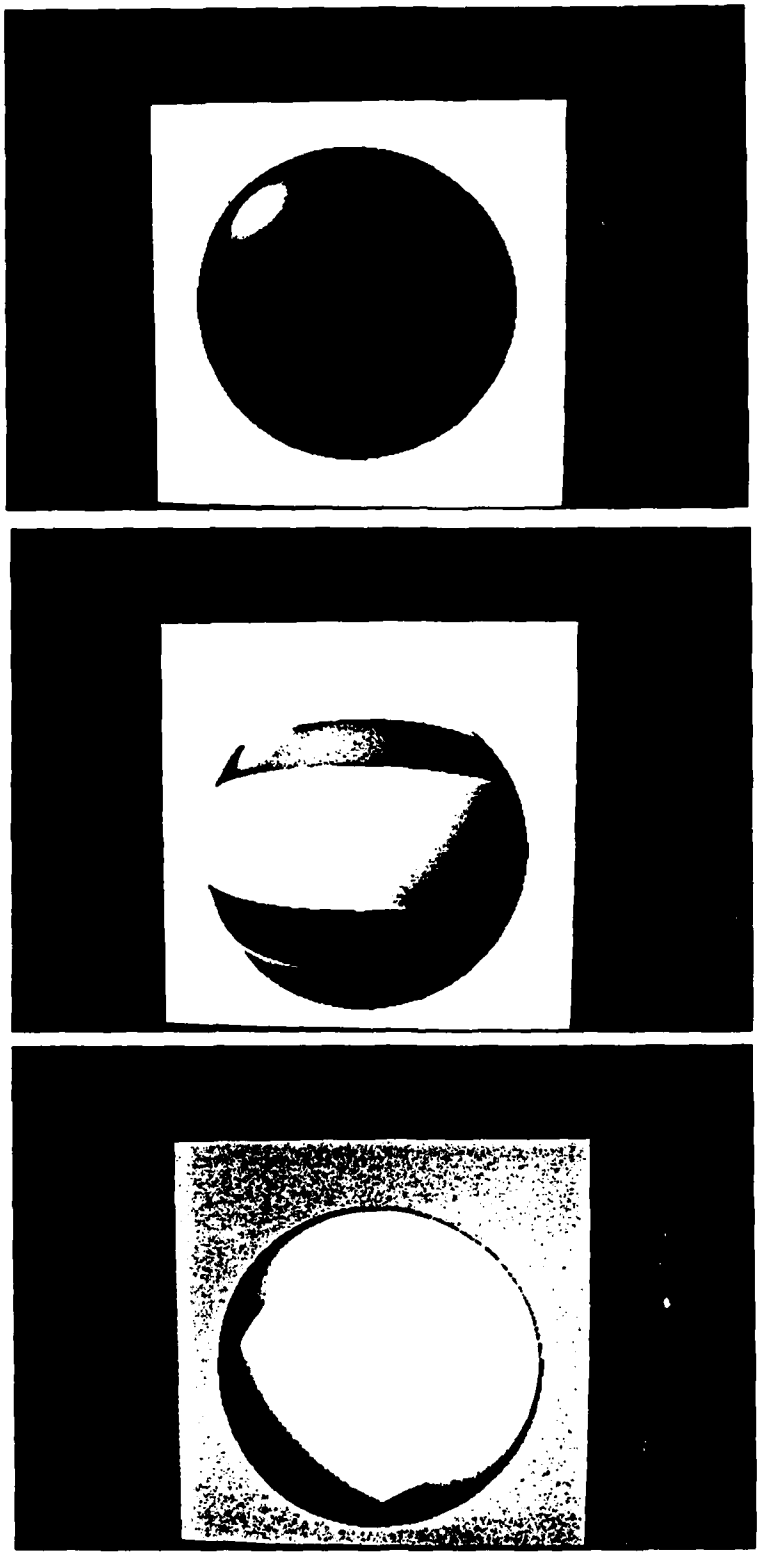


FIG. 3, CON'T.

The line of sight can be represented analytically by (eq.6).

$$x = (1-t)\gamma_1 + t\,\alpha_1 \qquad (6)$$
$$y = (1-t)\gamma_2 + t\,\alpha_2$$
$$z = (1-t)\gamma_3 + t\,\alpha_3.$$

(γ) is the location of the origin of the CAMERA coordinate system. (α) is a point on the image plane. The above equations are the general parametric form for a line through two points, α and γ, in terms of a parameter t.

A sphere can most conveniently be represented analytically in terms of polar coordinates.

<fig. 4>

In spherical coordinates:

$$x = r\sin\varphi\cos\theta \qquad (7)$$
$$y = r\sin\varphi\sin\theta$$
$$z = r\cos\varphi$$

$$\cos\varphi = z/r \qquad (8)$$
$$\tan\theta = y/x.$$

The equation for a sphere centered at the origin is simply

$$r^2 = x^2 + y^2 + z^2. \qquad (9)$$

r is, of course, the radius of the sphere.

<fig. 5>

In order to find the points of intersection between the line-of-sight and the sphere, the corresponding expressions for x, y, and z are set equal to each other. The angles φ and θ are eliminated using the triangles (fig.6).

<fig. 6>

Using these, all trigonometric expressions of φ and θ can be expresed in terms of x, y, z, and r.

It can now be found that:

$$x = K_1 z + K_2 \qquad (10)$$
$$y = K_3 z + K_4.$$

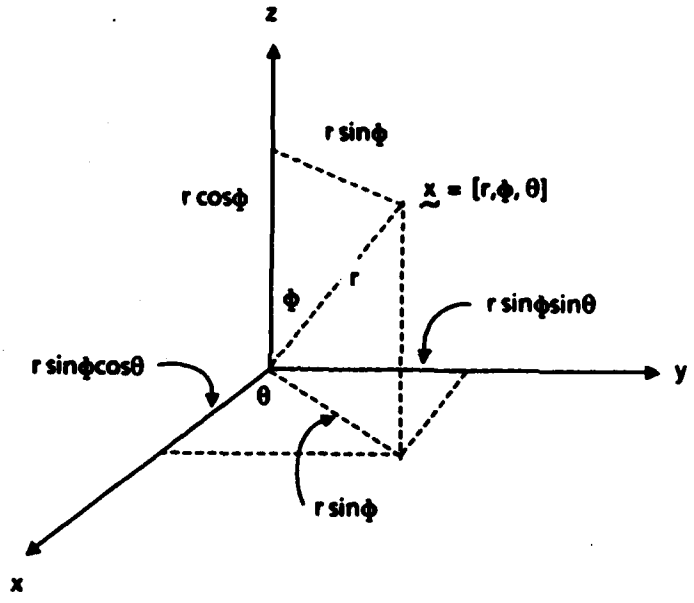


FIG. 4

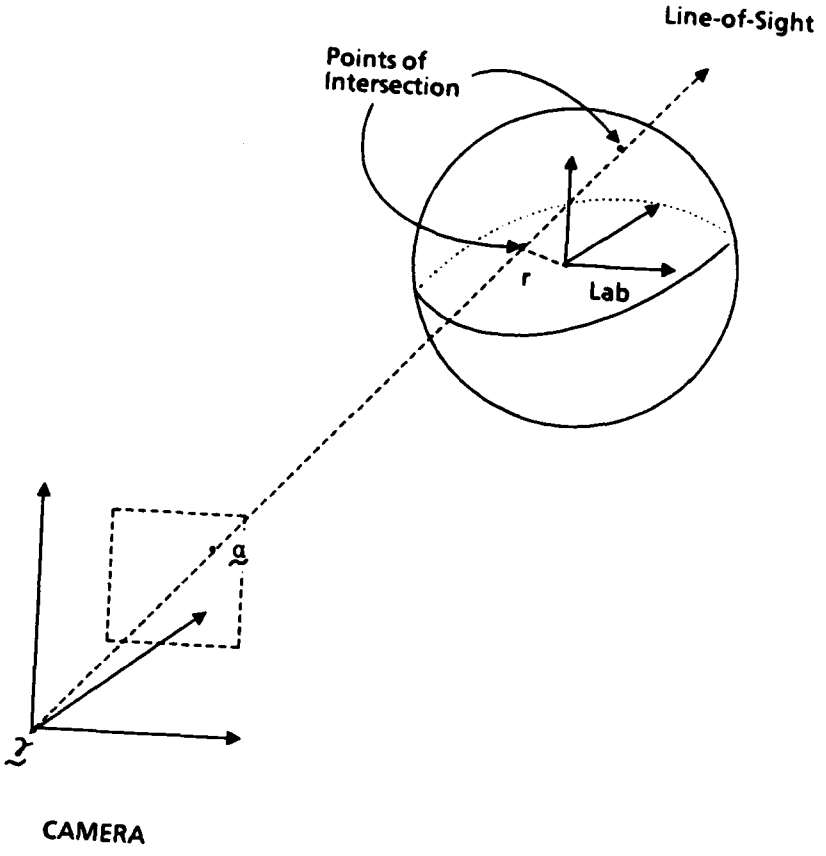


FIG. 5

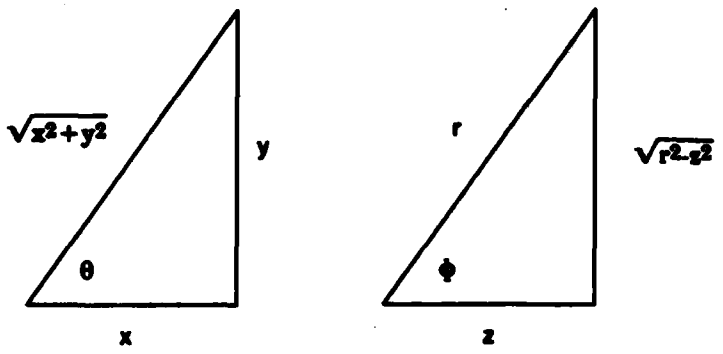


FIG. 6

where,

$$K_1 = (\alpha_1 - \gamma_1)/(\alpha_3 - \gamma_3) \qquad (11)$$
$$K_2 = (\gamma_1 \gamma_3 - \alpha_1 \gamma_3)/(\alpha_3 - \gamma_3) + \gamma_1$$
$$K_3 = (\alpha_2 - \gamma_2)/(\alpha_3 - \gamma_3)$$
$$K_4 = (\gamma_2 \gamma_3 - \alpha_2 \gamma_3)/(\alpha_3 - \gamma_3) + \gamma_2$$

Substituting (10) into (9):

$$r^2 = (K_1 z + K_2)^2 + (K_3 z + K_4)^2 + z^2, \qquad (12)$$

or:

$$r^2 = K_5 z^2 + K_6 z + K_7, \qquad (13)$$

where:

$$K_5 = K_1^2 + K_3^2 + 1 \qquad (14)$$
$$K_6 = 2K_1K_2 + 2K_3K_4$$
$$K_7 = K_2^2 + K_4^2.$$

A quadratic expression for z is now:

$$K_5 z^2 + K_6 z + K_8 = 0, \text{ where } K_8 = K_7 - r^2. \qquad (15)$$

Solving:

$$z = (-K_6 \pm \sqrt{K_6^2 - 4K_5K_8}) / 2K_5. \qquad (16)$$

The "$\pm$" term implies that there are two points of intersection. Of course, the nearer point of intersection must be chosen, the other one being invisible behind the sphere. When $K_6^2 = 4K_5K_8$, the $^+$ term vanishes and there is only one point of intersection. In this case, the line-of-sight grazes the sphere.

The ray tracing for the case of a cylinder is quite similar. Equation (9) is replaced by:

$$r^2 = x^2 + y^2, \text{ for } |y| < h/2, \text{ where h is the height of the cylinder.} \qquad (17)$$

A quadratic expression can now be found for the parameter t so that:

$$t = (-b \pm \sqrt{b^2 - 4ac}) / 2a \qquad (18)$$

And:

$$a = (\alpha_1 - \gamma_1)^2 + (\alpha_3 - \gamma_3)^2 \qquad (19)$$
$$b = 2(\gamma_1 (\alpha_1 - \gamma_1) + \gamma_3 (\alpha_3 - \gamma_3))$$
$$c = (\gamma_1^2 + \gamma_3^2 - r^2)$$

x, y, and z can be found as functions of t, α, and γ by (6). As before there will usually be two points of intersection, and the nearer one must be chosen.

Given two consecutive synthetic images for input, the spatial derivatives are found by subtracting the grey levels of adjacent points. The temporal derivatives for a given point are obtained by subtracting the grey level of the corresponding point in the subsequent image. Discrete differences in space and time must be used to approximate the infinitesimal "delta's" normally used in differentiation.

6. Experimental Results

Experiments were designed to address several questions. First, experiments were used to determine the magnitude of the rotation which would yield the most accurate results for the two statistical methods described above. Further experiments were used to draw some qualitative conclusions as to the effect that surface reflectance functions and surface patterns would have on the accuracy of the results. Finally, an experiment was designed to determine whether accurate results could be obtained when the motion consists of a combination of several simple rotations. As explained above, the inherent approximations used in the algorithm skew the camera coordinate system when two or more motions parameters are simultaneously non-zero.

6.1. Magnitude of Rotation

In order to determine the range of motion for which the egomotion algorithm would be most accurate an experiment was designed using the synthetic image shown in (fig. 7). This image consists of a sphere illuminated by a single light source positioned to the upper right of the sphere. The sphere has a partially specular reflectance function and several horizontal stripes across its surface.

In a series of tests, the A and C rotation parameters are set to zero and the value of the B parameter varied. This corresponds to a side-to-side "swivel" of the camera. The results are listed in Table 1 (Tables located at end of report), which summarizes the magnitude of the B rotation parameter, the number of pixels by which the object is displaced between successive images, and the results of the egomotion algorithm. The least-squared-error method works quite well for rotations small enough in magnitude that the object is displaced by less than an entire pixel between frames. In contrast, the Hough method yields its most accurate results for motions in which the object is shifted by a small number (1-3) of pixels between frames.

A complicating factor in this experiment is that the size of the Hough accumulator array is limited by the memory storage of the computer. For larger camera motions, larger incremental values are used between adjacent cells of the array so that the array can accomodate a larger range of motions. However, the size of the array increments effects the results yielded by the Hough Transformation. An experiment to illustrate this is summarized in Table 2.

In this experiment the same pair of spheres (of the type shown in fig. 7) are used in two different tests. In the first, adjacent cells of the accumulator array differ in value by 0.001. However, in the second adjacent cells differ by 0.0001. Clearly, the results of the algorithm are more nearly accurate for the case in which the accumulator

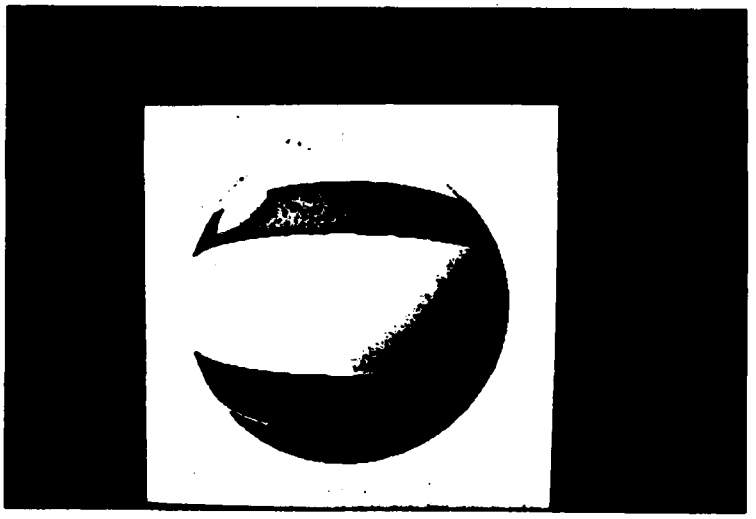
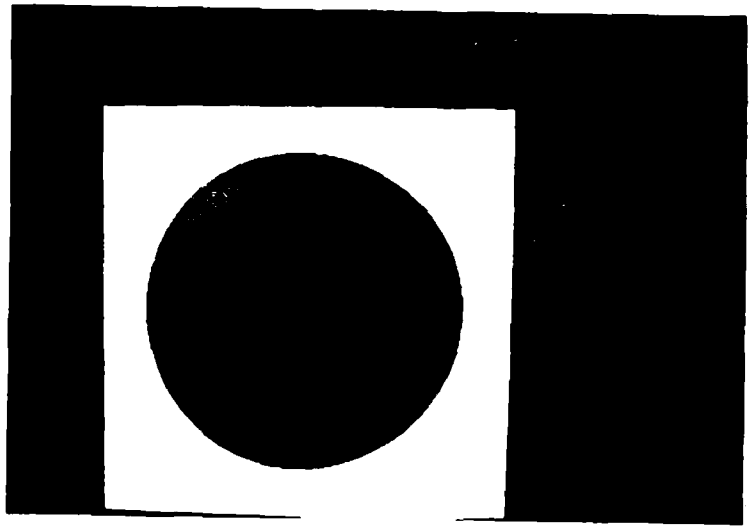
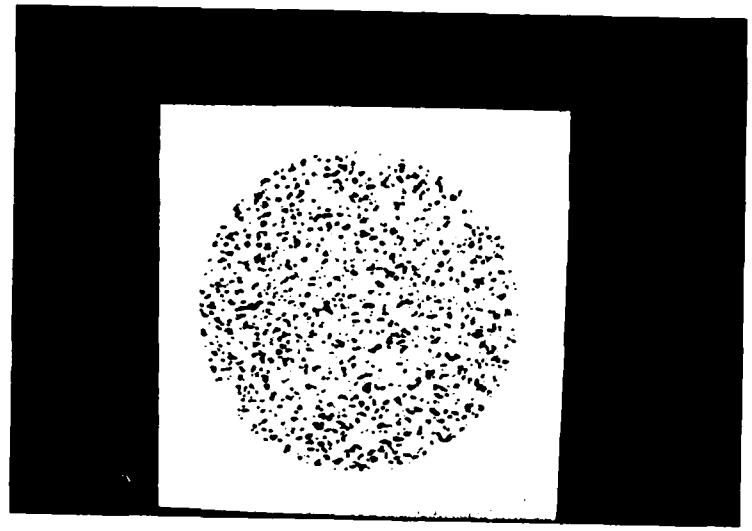


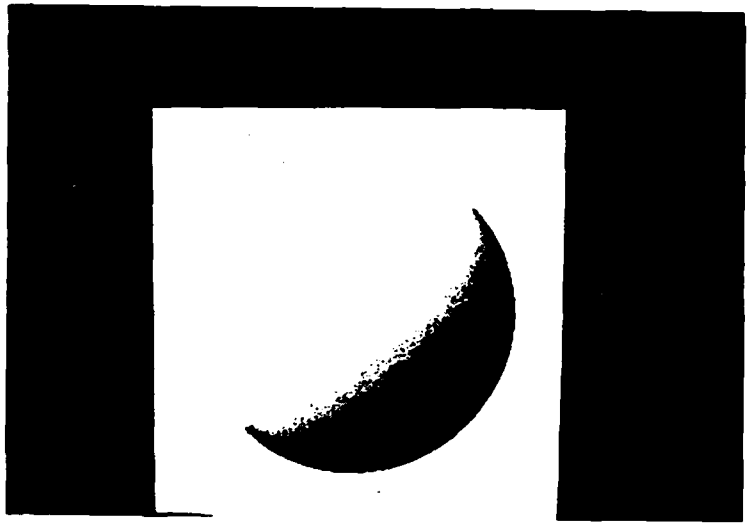
FIG. 7



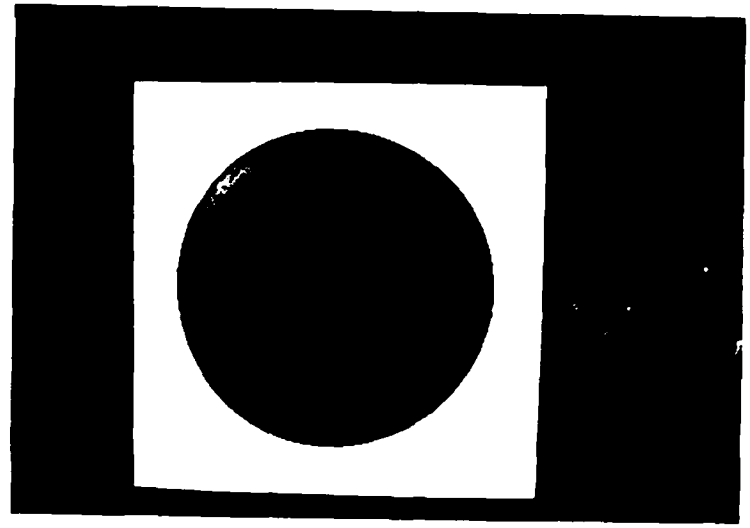
i. Lambertian



ii. Random



iii. Specular



iv. Specular x Random

FIG. 8

array provided finer resolution of values.

6.2. The Effect of Surface Pattern and Reflectance Function

It was determined above that accurate results can be obtained from the least-squared-error method when $A=C=0$ and $B=0.001$. This motion is repeated for a set of spheres whose surface patterns and reflectance functions differ. The spheres are pictured in (fig. 8) and the results of the experiments are summarized in Table 3. Accurate results are obtained when the surface of the sphere has a smoothly-varying intensity pattern which could be either a Lambertian or a specular reflectance function. Superimposed on this reflectance function can be a surface pattern which contains a few disjoint boundaries, such as the cases of the spheres having several horizontal stripes or the large "diamond" pattern. However, poor results are obtained when the sphere has a "random" surface pattern (lacking smooth variation between adjacent pixels). Poor results are also obtained when a random surface pattern is combined with a specular reflectance function (which restores some degree of order to the pixel intensity pattern).

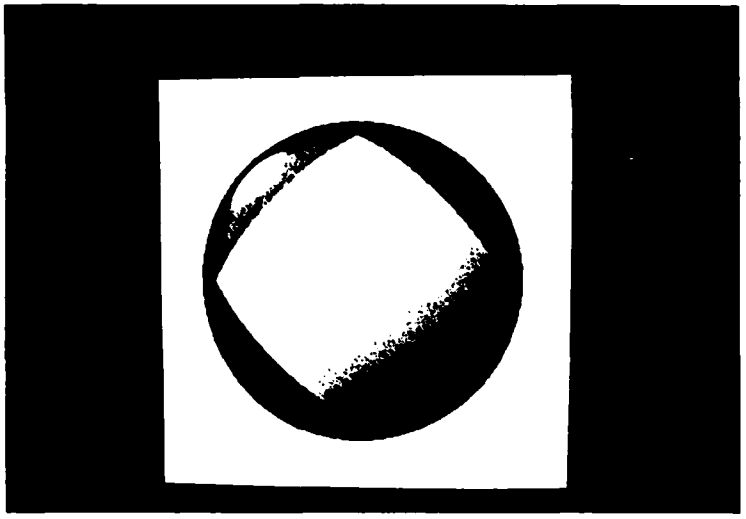
6.3. The Effect of Combined Motion

The sphere of (fig. 7) is again used to determine the effect of several combined motions on the algorithm. Motion parameters of small magnitude are used, and the least-squared-error method is employed here because these yielded the most accurate results in the first set of experiments. The results of several combinations of motion parameters are summarized in Table 4. For all of these experiments, the results are poor. One must conclude that the approximate transformation matrix T used to derive the Egomotion Equation, and also used to generate the synthetic graphics used as input, differs significantly enough from the exact transformation matrix T' to yield highly inaccurate results in the case when two or more motion parameters are combined. As a topic of future research, the Egomotion Equation could be rederived using the exact matrix T' instead of the approximate T.

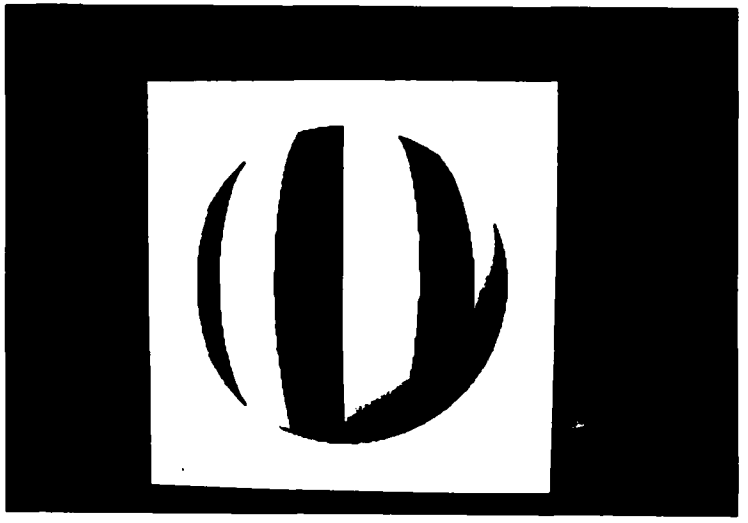
References

Aloimonos, J., and C.M. Brown. Direct Processing of Curvilinear Sensor Motion from a Sequence of Perspective Images. Working paper, Computer Science Dept., U. Rochester, February 1984.

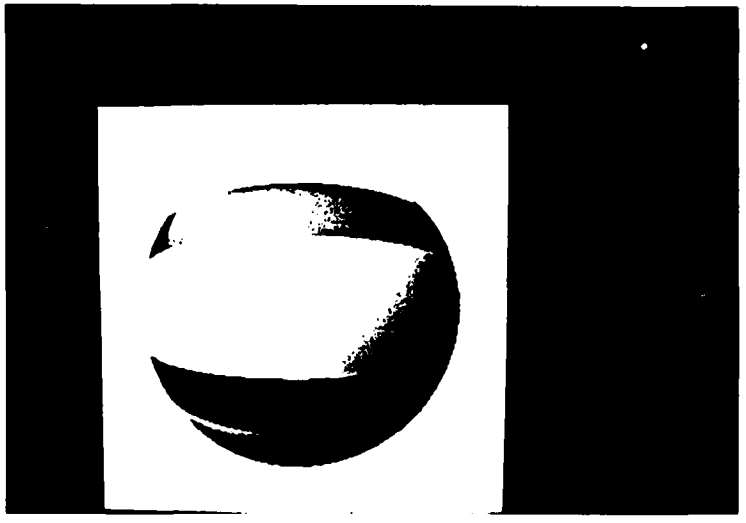
Giloi, W. *Interactive Computer Graphics.* Englewood Cliffs: Prentice-Hall, Inc., 1978.



v. Specular x Diamond



vi. Highly Specular x Vertical Stripes



vii. Specular x Stripes

FIG. 8, CON'T.

TABLE 1

A = C = 0

B	Number of Pixels Displaced	Least-Squared-Error Results			Hough Transformation Results			Size of Hough Accumulator Array Increments	# of Votes
		A	B	C	A	B	C		
0.0005	0	0.000129	0.001273	-0.000157	-0.0008	0.0022	0.0004	0.0001	7
0.001	0	0.000092	0.001932	-0.000790	-0.0006	0.0012	-0.0009	0.0001	5
					0.0000	0.0013	-0.0020		
					0.0000	0.0022	0.0007		
					0.0001	0.0028	0.0020		
					0.0001	0.0029	0.0023		
					0.0004	0.0017	0.0011		
					0.0005	0.0018	0.0005		
					0.0005	0.0032	0.0025		
					0.0007	0.0018	0.0004		
					0.0007	0.0031	0.0009		
					0.0007	0.0034	0.0025		
					0.0013	0.0012	0.0025		
					0.0013	0.0013	0.0021		
					0.0014	0.0012	0.0025		
					0.0014	0.0013	0.0021		
					0.0015	0.0014	0.0016		
					0.0021	0.0011	0.0025		
0.002	0	0.000038	0.002565	-0.000966	0.0001	0.0028	0.0020	0.0001	6
0.004	1	0.000060	0.005192	-0.001896	0.0002	0.0055	0.0037	0.0001	12
					0.0002	0.0057	0.0045		
					0.0038	0.0053	-0.0050		
0.008	2	-0.000420	0.065875	-0.001854	0.000	0.010	0.000	0.001	14
0.016	3	-0.001051	0.006744	-0.001461	0.000	0.017	0.012	0.001	9
					0.006	0.014	0.000		
					0.016	0.005	-0.006		
0.032	6	-0.001586	0.007677	-0.000955	0.01	0.022	-0.02	0.01	16
0.064	12	-0.001871	0.008274	-0.000425	0.02	0.054	-0.04	0.01	14
0.128	22	-0.001998	0.008861	0.000063	0.12	0.038	-0.08	0.01	10
0.256	40	-0.001827	0.009059	-0.000737	0.18	0.076	-0.06	0.01	8

TABLE 2

A = C = 0
B = 0.004

Size of Hough Accumulator Array Increments	Hough Transformation Results			Number of Votes
	A	B	C	
0.001	0.002	0.008	-0.001	12
0.0001	0.0002	0.0055	0.0037	6
	0.0002	0.0057	0.0045	
	0.0038	0.0053	-0.0050	

TABLE 3

A = C = 0
0.001

	Sphere	Least-Squared-Error Results		
		A	B	C
i.	Lambertian	-0.000048	0.001025	-0.000019
ii.	Random	-0.002594	0.002572	-0.000088
iii.	Specular	-0.000191	0.001205	0.000384
iv.	Specular x Random	-0.002588	0.002220	-0.001412
v.	Specular x Diamond	0.000009	0.000856	-0.000580
vi.	Highly Specular x Vertical Stripes	-0.000970	0.000991	-0.000262
vii.	Specular x Stripes	0.000092	0.001932	-0.000790

TABLE 4

Motion Parameters			Least-Squared-Error Results		
A	B	C	A	B	C
0.001	0.001	0	0.000096	0.001755	0.000191
0	0.001	0.001	-0.000054	0.001059	0.000449
0.001	0.001	0.001	0.000079	0.001789	0.000668
0.0005	0.001	0	0.000059	0.001381	0.000054
0.0007	0.0007	0	0.000071	0.001256	0.000107
0.0008	0.0011	0.0006	0.000071	0.001727	0.000421

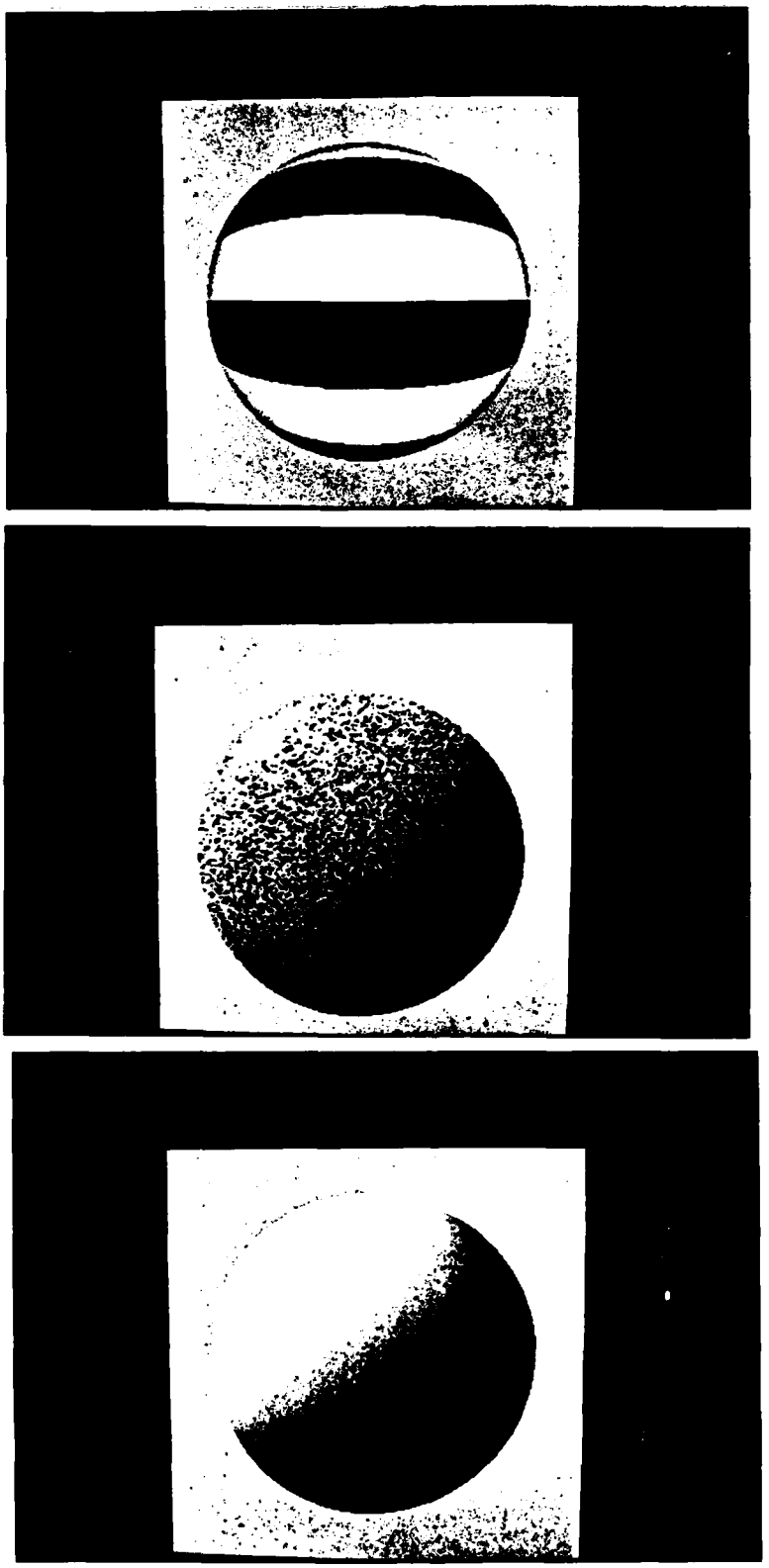


FIG. 3, CON'T.

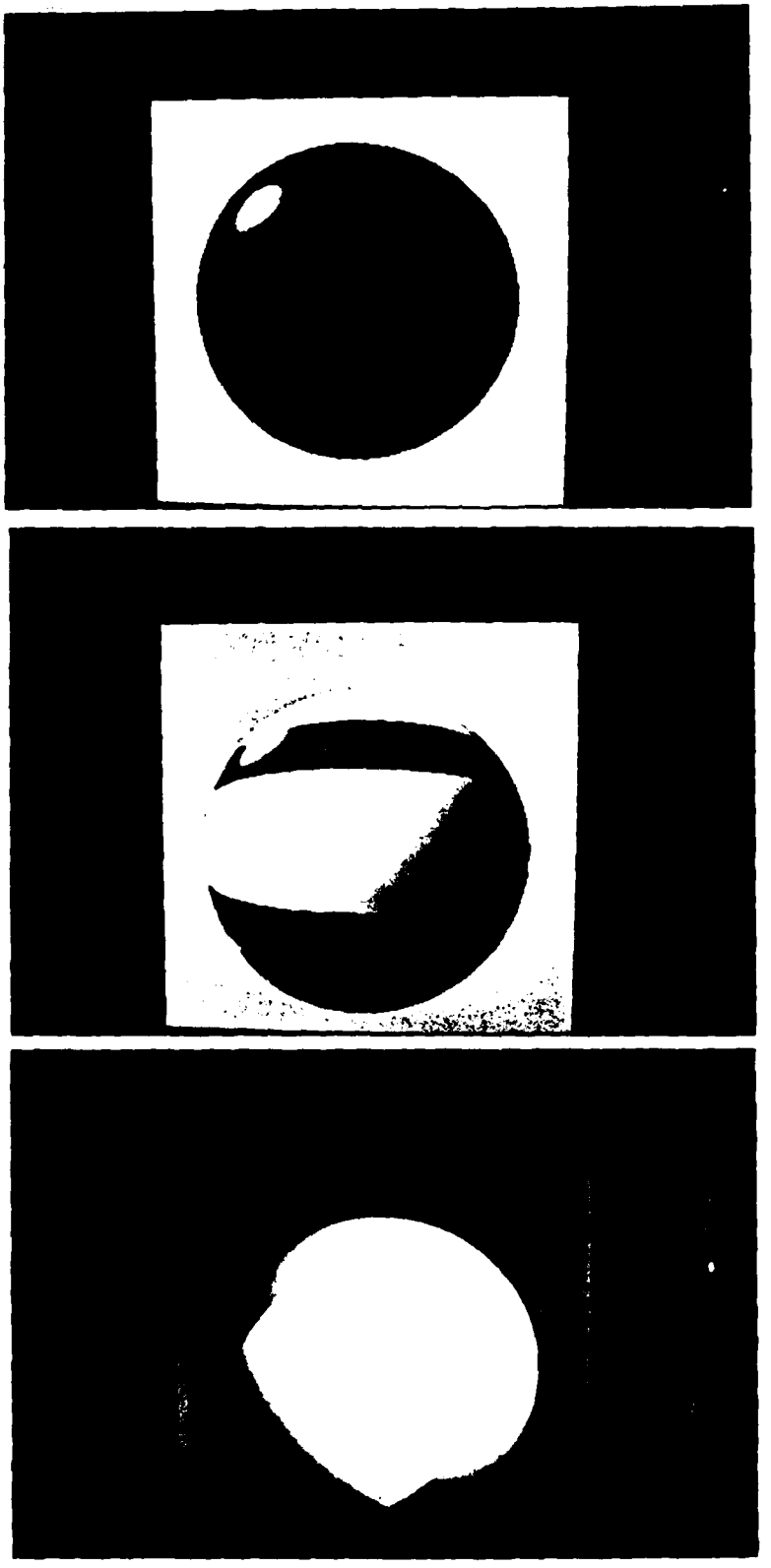


FIG. 3, CON'T.

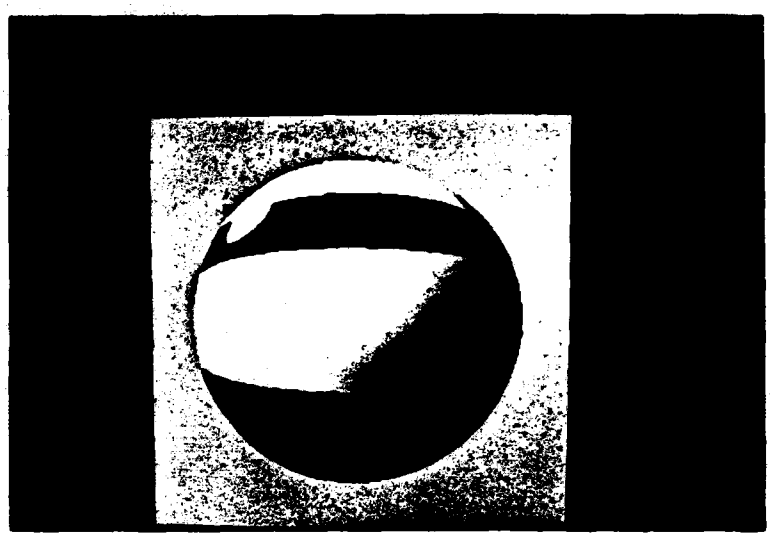
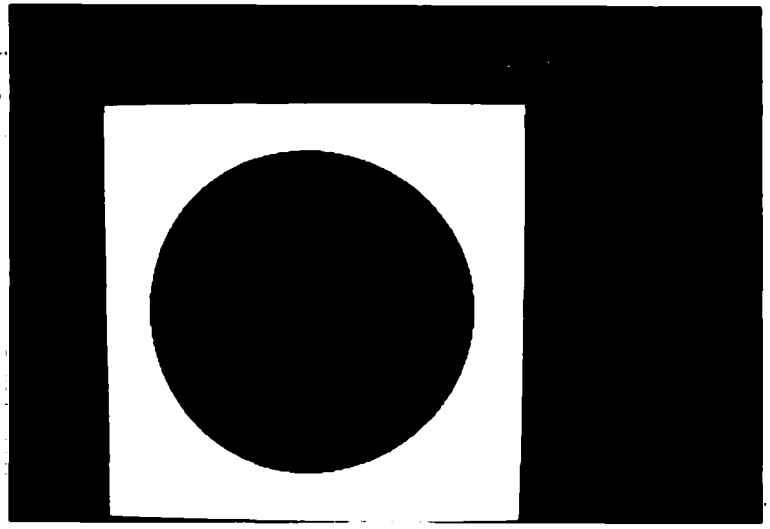
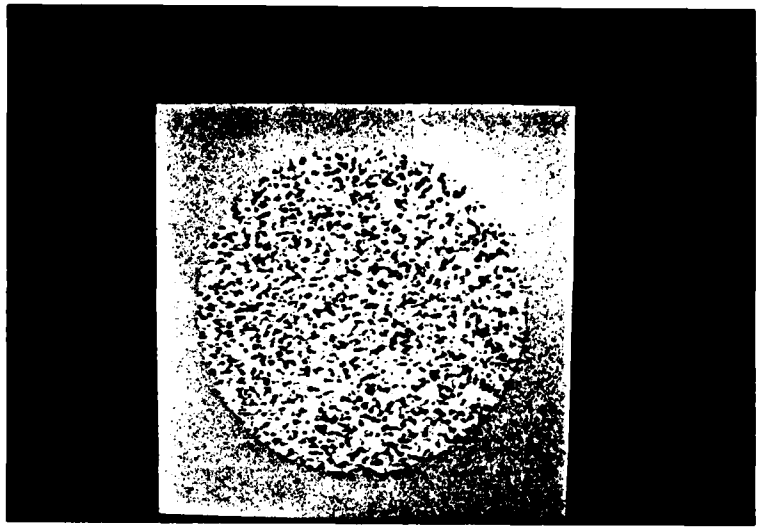


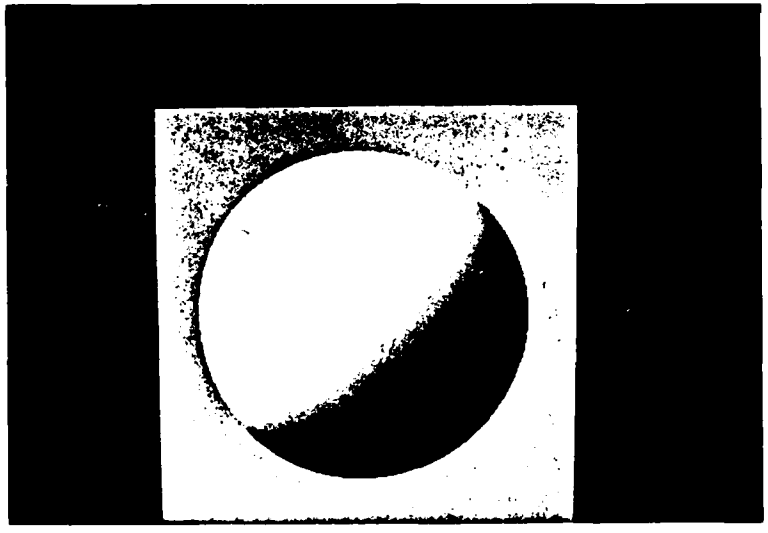
FIG. 7



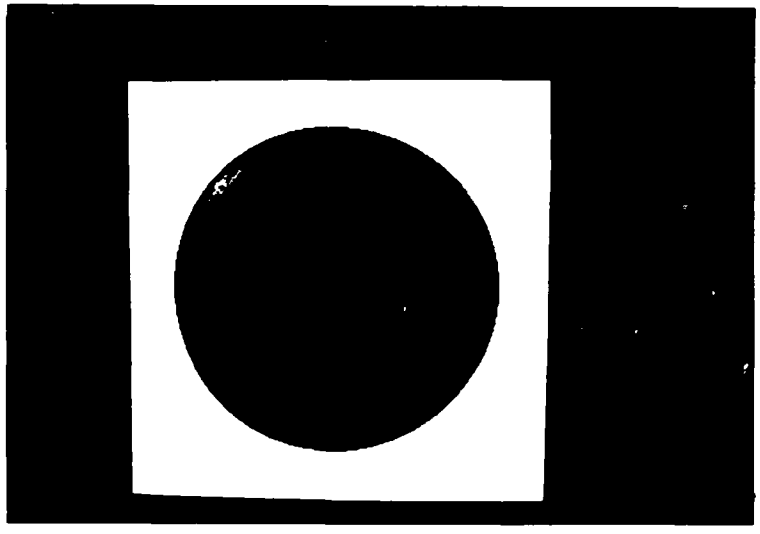
i. Lambertian



ii. Random

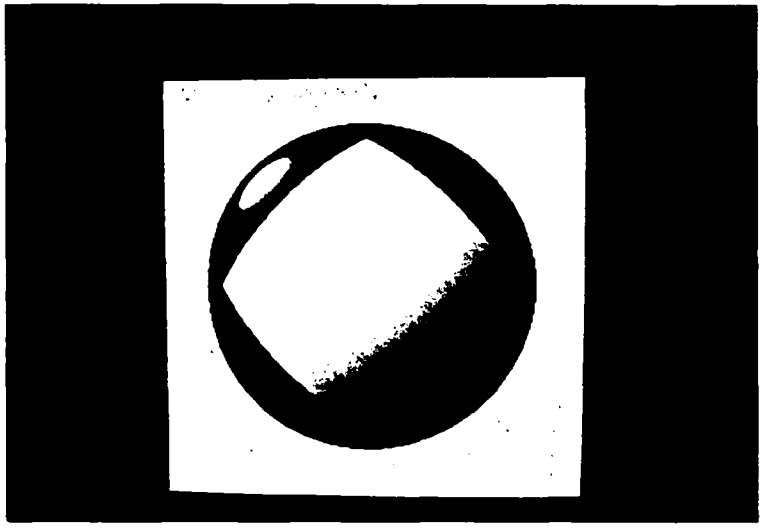


iii. Specular

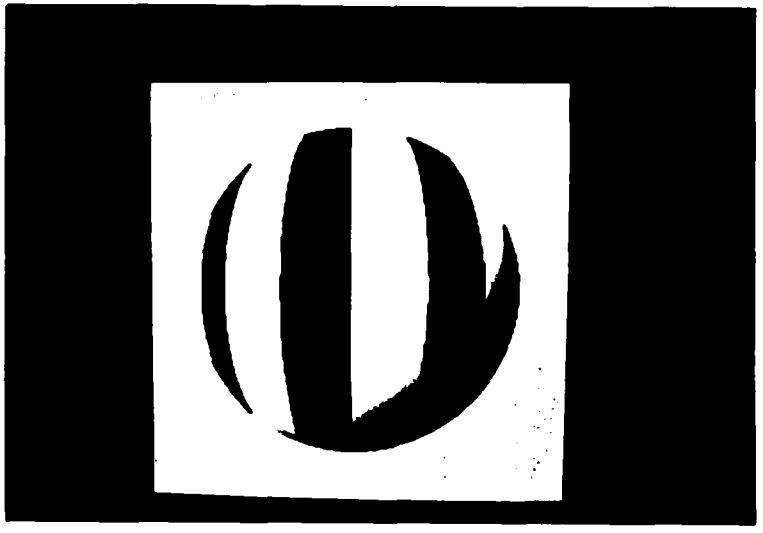


iv. Specular x Random

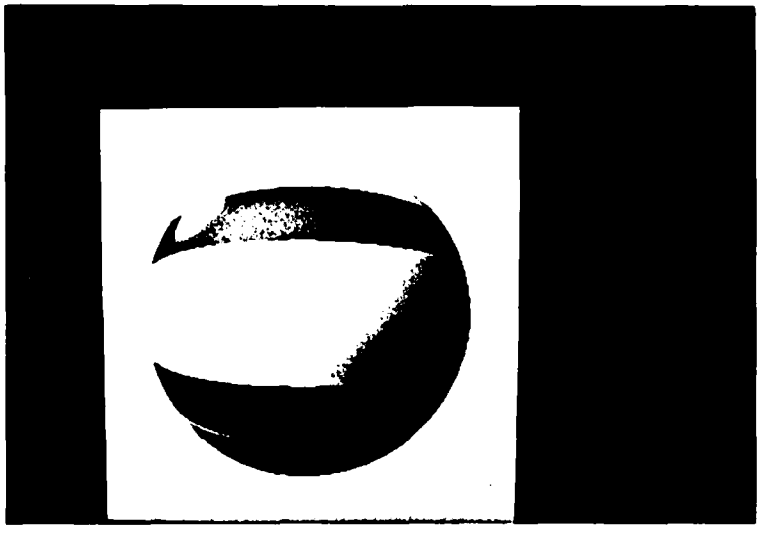
FIG. 8



v. Specular x Diamond



vi. Highly Specular x Vertical Stripes



vii. Specular x Stripes

FIG. 8, CON'T.

A170 882

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 152	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Experiments in Rotational Egomotion Calculation		5. TYPE OF REPORT & PERIOD COVERED Technical REport 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Brion Sarachan		8. CONTRACT OR GRANT NUMBER(s) DACA76-85-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science University of Rochester Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA/ 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE February 1985 13. NUMBER OF PAGES 17
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

passive navigation, optic flow, egomotion, motion vision, synthetic images

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The motion of an observer relative to a fixed environment can be determined from a sequence of successive perspective images. The method is based on constraints between the motion parameters, the image intensities, and the shape of the body in view. A method is developed for experimentally verifying the algorithm using synthetic input for the case of rotational motion.

Technical problems arise in the representation of continuous mathematics in the discrete domain of a computer system. The effects of some of these difficulties are examined analytically and experimentally.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

END

DATE
FILMED

9-86